

# Postproceso estadístico de modelos con *machine learning*: una aplicación al $\gamma$ SREPS y al Harmonie

<https://doi.org/10.31978/639-19-010-0.139>

David Quintero Plaza<sup>1</sup> ([dquinterop@aemet.es](mailto:dquinterop@aemet.es))

<sup>1</sup>AEMET / Delegación Territorial en Canarias

## RESUMEN

Las técnicas de *machine learning* o aprendizaje automático, provenientes del campo de la Inteligencia Artificial, están demostrando ser extraordinariamente exitosas para extraer significado de un conjunto de datos. En este trabajo se ha buscado aplicar las técnicas de *machine learning* al postproceso estadístico de dos modelos: el ensemble  $\gamma$ SREPS de AEMET y el Harmonie v40. Para el caso del  $\gamma$ SREPS se ha hecho una calibración determinista, miembro a miembro, de la temperatura a 2 metros, velocidad del viento a 10 metros y precipitación en 24 horas en varios aeropuertos de España, que representan diferentes condiciones climáticas. Para el Harmonie v40 el objetivo ha sido mejorar la predicción vectorial del viento a 10 metros en el aeropuerto de Tenerife Sur.

Los resultados han sido muy buenos con el viento y la temperatura. Para la precipitación existe, como era de esperar, algo más de ambigüedad.

**PALABRAS CLAVE:** postproceso; estadística; calibración; *machine learning*; modelos meteorológicos; aplicaciones.

## 1. INTRODUCCIÓN

Las técnicas de *machine learning* se cuentan entre las más exitosas para analizar, clasificar y, en definitiva, extraer significado del océano de datos en el que vivimos. Su éxito ha sido tal que pueden verse noticias sobre ellas no solo en revistas especializadas sino incluso en prensa generalista (EL PAÍS, 2016).

En las ciencias atmosféricas se han ido aplicando técnicas de *machine learning* desde hace tiempo, para la mejora o calibración de la predicción, para la climatología, para la teledetección y más (HSIEH, 2009). Pero los éxitos recientes de redes neuronales profundas (*deep learning*) han disparado la atención hacia estos métodos y ahora toda una generación de expertos en las ciencias atmosféricas han añadido la *machine learning* como una herramienta más en su trabajo (JONES, 2017).

Nuestro trabajo se centra en la calibración determinista de modelos, en concreto de los miembros del  $\gamma$ SREPS y del modelo Harmonie. Primero hablaremos muy someramente de en qué consiste la *machine learning* y después pasaremos a explicar nuestro trabajo y sus conclusiones.

## 2. BREVE PANORAMA DE LA MACHINE LEARNING

La *machine learning* (en adelante, ML) puede verse como estadística aplicada, en la que el énfasis se hace más en la potencia de cálculo y en el pragmatismo que en la demostración rigurosa de determinados resultados. Muchos de los términos son intercambiables: *learning*, por ejemplo, sería «inferencia» en estadística. La ML puede dividirse en tres grandes apartados: aprendizaje supervisado, no supervisado y aprendizaje con refuerzo, dependiendo de que tengamos unos datos objetivos a los que acercarnos, no los tengamos pero podamos evaluar la calidad del aprendizaje de alguna forma o no tengamos ni siquiera los datos, sino un sistema de prueba y error. En nuestro trabajo, dado que consideramos las observaciones como la «realidad» que queremos simular, estamos en el paradigma supervisado.

Dentro del aprendizaje supervisado tenemos problemas de clasificación o de regresión, dependiendo de que nuestros datos sean discretos o continuos. En el caso que nos ocupa, en el que, como hemos dicho en el resumen, vamos a calibrar temperatura a 2 metros, viento a 10 metros y precipitación en 24 horas, todas ellas son variables continuas, así que haremos regresión.

Nuestro objetivo principal va a ser ajustar un conjunto de datos tanto como podamos pero sin llegar a hacerlo perfecto. Puede parecer paradójico, pero los algoritmos de ML son tan potentes, que son perfectamente capaces de memorizar un conjunto de datos, clavando el aprendizaje. Como queremos que nuestros algoritmos generalicen bien el caso de datos no observados, en otras palabras, como queremos estimar realmente los parámetros poblacionales, no los muestrales, debemos luchar contra esta perfección. A este ajuste perfecto se le conoce como *overfitting* o sobreajuste. Ya John Von Neumann advirtió irónicamente contra esta tendencia: «con cuatro parámetros libres ajusto tus datos para que tengan forma de elefante, y con cinco hago que mueva su trompa».

El conjunto de algoritmos de ML es muy rico y variado, y nosotros hemos hecho pruebas con muchos de ellos. A continuación se dará una brevísima descripción de las ideas fundamentales tras los algoritmos que hemos usado.

### 2.1. Ridge, Lasso, Elastic Net y Bayesian Ridge

Estos algoritmos son generalizaciones de la regresión lineal ordinaria por mínimos cuadrados, minimizando el error cuadrático medio (MSE). La diferencia es que llevan un término extra que es proporcional a los coeficientes de la recta de regresión, penalizando valores elevados que lleven a sobreajustes. Si los coeficientes de la regresión entran con  $p = 2$ , tenemos *Ridge*; con  $p = 1$  tenemos *Lasso*. *Elastic Net* sería una combinación lineal de las dos posibilidades anteriores. Podemos ver esto en la figura 1.

$$MSE = \sum_{i=1}^n \left( y_i - \sum_{j=1}^p X_{ij} \beta_j \right)^2 + \lambda \sum_{j=1}^p |\beta_j|^p$$

Figura 1. Ecuación de minimización.  $X$  es la matriz de predictores,  $y$  las observaciones,  $\beta$  los coeficientes a averiguar y  $\lambda$  el «peso» del término de penalizaciones.

### 2.2. Random Forests y Boostings

Un *random forest* consiste en la creación de un árbol de decisión: la figura familiar de un conjunto de nodos y ramas que son funciones de los datos. La función de los datos será en nuestro caso de nuevo el error cuadrático medio. Un conjunto de árboles de decisión forman un bosque o *forest*. De este conjunto se extraen valores de los parámetros a estimar.

Las técnicas de *boosting* combinan los *random forests* con la minimización de una función, en este caso el error cuadrático medio. La minimización sigue el proceso de «descenso del gradiente». Si sabemos que el gradiente da la dirección de máximo crecimiento de una función de varias variables, podemos movernos en la dirección contraria para encontrar el máximo decrecimiento (véase la figura 2).

$$w_{i+1} = w_i - \eta \nabla g(w)$$

Figura 2. Descenso del gradiente de una función.  $\eta$  es el *step* o paso entre dos iteraciones. El parámetro es  $w$  y la función a minimizar es  $g(w)$ .

En ocasiones se añade a la ecuación de la figura 2 algún término extra proporcional a la iteración anterior del proceso, para que el movimiento del descenso no sea «zigzagueante», sino más directo.

### 2.3. SVR y redes neuronales

La regresión mediante vectores singulares (SVR) es matemáticamente sutil y llevaría tiempo de explicar: digamos que la idea clave es ir a espacios de dimensión superior en los que es más fácil resolver un determinado problema numérico. Para evitar que el coste computacional de trabajar en estos espacios se dispare, se utilizan unas funciones llamadas *kernels* que ahorran mucho cálculo, ya que realizan el mapeo al espacio de dimensión superior y el cálculo de la distancia entre puntos simultáneamente. Un *kernel* muy habitual es una función de tipo gaussiano.

La red neuronal es una imitación del cerebro humano, con capas de neuronas, que son unidades que realizan operaciones de suma pesada sobre los datos de entrada seguida de una operación no lineal. Cada neurona tiene unos coeficientes que hay que ajustar mediante un proceso llamado *backpropagation*, que es básicamente una minimización de una función por descenso del gradiente. Las redes neuronales vienen en multitud de diseños y son las responsables del gran éxito de la ML.

Existen múltiples referencias para profundizar en el fascinante mundo de la ML. Creemos que (HSIEH, 2009) es muy completo y a la vez aplicado al campo de la meteorología.

## 3. APLICACIONES

En nuestro trabajo hemos calibrado con ML dos modelos: el  $\gamma$ SREPS y el Harmonie v40. Aunque ambas calibraciones tienen muchos elementos en común, las trataremos por separado.

### 3.1. Harmonie v40 para el aeropuerto de Tenerife Sur - Reina Sofía

Es sabido que el aeropuerto de Tenerife Sur - Reina Sofía es el que tiene los peores *scores* de verificación del viento de los TAF de toda España. Esto no debe resultar sorprendente, pues incluso los modelos de alta resolución resultan limitados para un aeropuerto ubicado en un lugar tan singular. En la figura 3 podemos apreciar que el aeropuerto se ubica en el extremo sur de la isla de Tenerife, un punto de confluencia de flujos del oeste y del noreste. Además, su cercanía a la costa hace que en ocasiones en que los vientos alisios no son tan importantes, dominen los fenómenos de brisas. Por si fuera poco, la presencia del Teide justo encima, perturbando los flujos, y provocando rápidos cambios del terreno que los modelos a 2,5 kilómetros de resolución no captan del todo bien.

Para abordar el problema, lo primero que tuvimos que hacer fue una transformación de los vectores del viento, ya que las salidas de Harmonie están en una proyección Lambert cónica conforme (LCC) y nosotros necesitamos una cartesiana tipo rosa de los vientos, que es la proyección en la que están las observaciones de



Figura 3. El aeropuerto de Tenerife Sur según Google Maps. La flecha azul representa el flujo, normalmente intenso, del alisio. Las flechas rojas son brisas costeras y la ocasional brisa del suroeste en el caso de borrascas al oeste de Canarias. Las flechas negras hacen referencia a la débil contrabrisa nocturna cuando el alisio no es importante.

los partes METAR de los aeropuertos. Esta transformación se hizo en dos etapas: una rotación de LCC a cartesiano mediante una matriz jacobiana y un desplazamiento para obtener la rosa de los vientos. Estamos pues, en el caso de una transformación afín.

Después buscamos los cuatro puntos más cercanos al punto de observación, esto es, el punto donde se toman los datos con los que se elabora el METAR. Para ello hicimos uso de la librería *ecCodes* del ECMWF para manejar ficheros GRIB. Estos puntos se cargaron en una estructura de datos llamada *dataframe*, de la librería de software *pandas*, de Python. Esta estructura de datos es ideal para tabular y organizar grandes cantidades de datos. Comparábamos datos de viento de las componentes  $u$  y  $v$  a 10 metros, procedentes del METAR, con las  $u$  y  $v$  de los 4 puntos más cercanos. Además, incluimos valores de temperatura a 2 metros y presión en superficie de estos 4 puntos más cercanos para añadir información extra a la regresión (regresión multivariable).

Se hizo un control de calidad para evitar *outliers*. Esto es siempre algo sutil y problemático. Para no acabar en un autoengaño, solo los errores gruesos de modelo y observación se descartaron. Esto es, temperaturas superiores en módulo a  $80\text{ }^{\circ}\text{C}$ , componentes de viento superiores a  $100\text{ m/s}$  y presiones superiores a  $1200\text{ hPa}$  e inferiores a  $800\text{ hPa}$ . Somos conscientes de que es posible que aun así hayan entrado valores no válidos, pero no se pueden eliminar estos sin quitar valores buenos o valores que penalicen y encaucen el entrenamiento.

El periodo de entrenamiento fue de julio de 2016 a enero de 2018. El intervalo horario que se postprocesó fue desde el H+06 del modelo hasta el H+36. El METAR de Tenerife Sur se tiene cada media hora y el aeropuerto es un H24. Nótese un detalle importante: se postprocesaron con el mismo peso todas las horas de una pasada del modelo; es decir, tuvieron el mismo peso H+06 que H+07 o que incluso H+33. Esto no es la forma más óptima de hacerlo, nos movieron motivos prácticos: un procesado horario hubiera reducido considerablemente el tamaño de la base de datos de entrenamiento, factor muy importante en la ML. Sin duda, el trabajo futuro, ahora que empezamos a tener una base de datos de tamaño decente, lo haremos con un entrenamiento propio para cada hora.

En las figuras 4 y 5 se muestran las gráficas de cada entrenamiento para las componentes  $u$  y  $v$  del viento. Nótese que seguimos la técnica habitual en ML de no presentar un único resultado, un único entrenamiento,

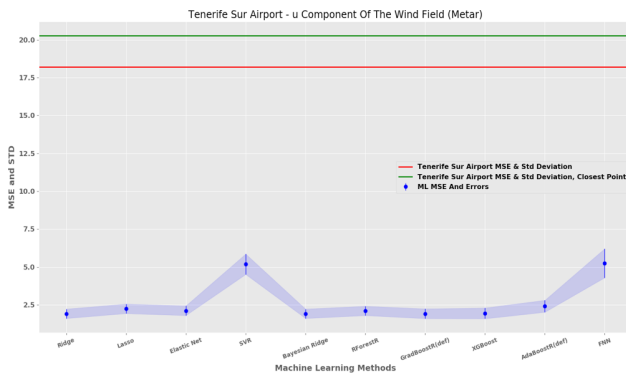


Figura 4. Componente  $u$  del viento a 10 metros para Tenerife Sur, modelo Harmonie v40, comparación de con entrenamiento y sin entrenamiento; en azul la dispersión obtenida tras hacer validación cruzada.

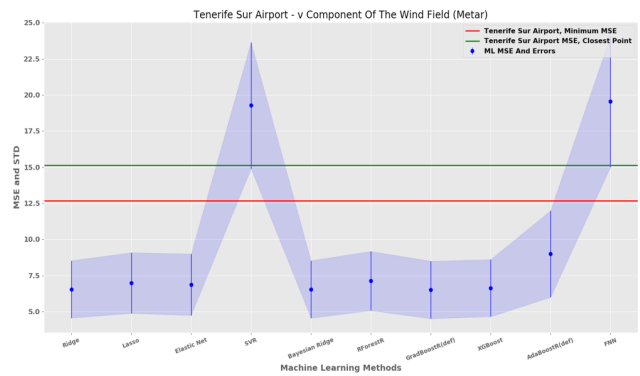


Figura 5. Componente  $v$  del viento a 10 metros para Tenerife Sur, modelo Harmonie v40, comparación de con entrenamiento y sin entrenamiento; en azul la dispersión obtenida tras hacer validación cruzada.

sino un *ensemble* de entrenamientos sobre diferentes particiones de la base de datos de entrenamiento. A este conjunto de entrenamientos se le calcula la media y la desviación estándar. Este procedimiento se conoce como *cross-validation* o validación cruzada, y es una medida más fiable que presentar un único entrenamiento.

### 3.2. $\gamma$ SREPS en diferentes aeropuertos de España

El trabajo con el modelo  $\gamma$ SREPS ha seguido las mismas ideas que con el Harmonie v40 y por tanto, para evitar repeticiones innecesarias, nos centraremos aquí en lo que es sustancialmente diferente. La principal novedad es que el campo de presión en superficie no se utilizó, a diferencia del Harmonie v40. Se pretende incluir en futuras implementaciones.

El trabajo con el viento comenzó calibrando únicamente la velocidad, aunque de ahora en adelante se hará también componente a componente. Los resultados fueron especialmente buenos con prácticamente todos los métodos ML.

En el caso de la temperatura se tuvo en cuenta que ningún punto vecino seleccionado fuera de mar, dado que de los 5 aeropuertos de referencia con los que se trabajó, 3 de ellos estaban próximos a la costa. Esto se hizo mediante una máscara tierra-mar, para evitar la diferente onda térmica que los puntos de mar tienen frente a los de tierra (es cierto que los algoritmos pueden corregir muy bien este error sistemático, pero la *feature engineering* de la ML consiste en aplicar el máximo conocimiento físico posible a un sistema, esto es, en facilitarle lo máximo posible las cosas a los algoritmos). Los métodos fueron muy bien, quizá no tan bien como el viento a 10 metros, pero esto puede explicarse en gran medida sabiendo que en el viento el algoritmo tiene mayor margen de mejora, a diferencia de en la temperatura.

Por último, respecto a la precipitación en 24 horas, la novedad fue utilizar muchos más vecinos para tener un área mayor y más representativa; en concreto se tomaron los 12 vecinos más cercanos al punto de observación. Esto permite tratar la problemática del *double penalty*, que no solo aparece en la verificación sino también en calibración. Otra cuestión importante es que para la precipitación ayuda considerablemente el llevar a cabo una *estandarización* de los datos de entrada o predictores, esto es, convertir los valores de cada variable a los de una distribución normal de media 0 y varianza 1. Esta práctica es habitual en la literatura de ML; si no se hizo para la temperatura ni el viento es porque no se vio que fuera necesaria. Mostramos en la figura 7 un ejemplo de calibración de la precipitación en 24 horas. Las calibraciones de la precipitación son especialmente difíciles, como por otro lado era de esperar. Creemos que todavía habría mucho trabajo que hacer para esta variable.

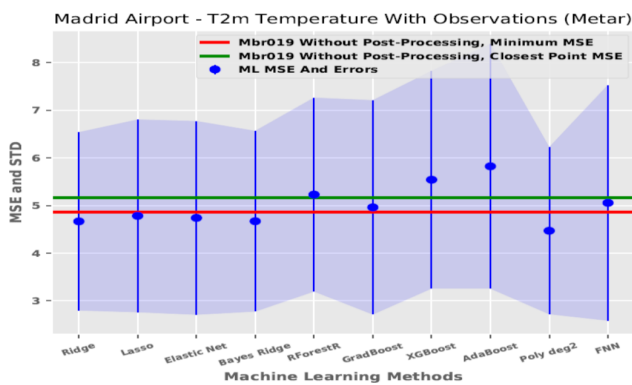


Figura 6. Temperatura a 2 metros en el aeropuerto de Madrid-Barajas – Adolfo Suárez para el miembro 19 del  $\gamma$ SREPS, el modelo WRF con las condiciones de contorno del modelo global de la Japan Meteorological Agency.

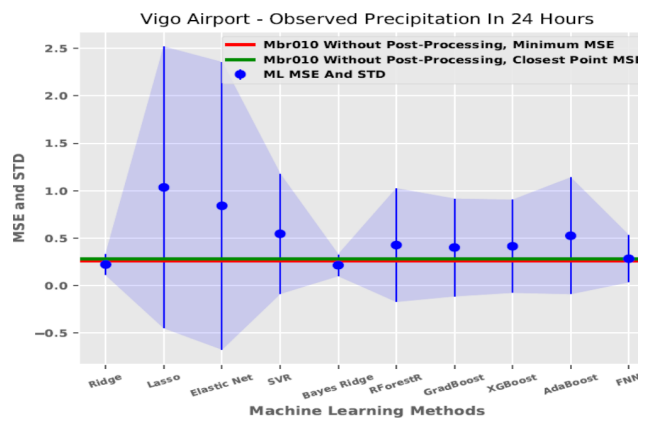


Figura 7. Precipitación en 24 horas en el aeropuerto de Vigo según el miembro 10 del  $\gamma$ SREPS, esto es, el modelo Harmonie-Alaro con las condiciones de contorno del modelo global de Météo-France.

#### 4. CONCLUSIONES

Creemos que este trabajo muestra el poder y la importancia de la ML para calibrar modelos meteorológicos. Como era de esperar, la calibración es exitosa en los casos de la temperatura y el viento y aporta algo en el caso de la precipitación, aunque creemos que en el caso de esta variable se precisa mucho más trabajo y estudio.

Lo hecho hasta ahora es tan solo un primer paso de la aplicación de calibración de modelos mediante ML. Creemos que las aplicaciones futuras son tan prometedoras como los resultados actuales. Con el tiempo, quizá podría extenderse la calibración no solo a aeropuertos sino a ciudades, a más variables además de las tres citadas e incluso no solo a predicciones de punto sino de áreas extensas.

#### REFERENCIAS

- EL PAÍS (JAVIER SALAS), 2016. [https://elpais.com/elpais/2016/01/26/ciencia/1453766578\\_683799.html](https://elpais.com/elpais/2016/01/26/ciencia/1453766578_683799.html).
- HSIEH, W., 2009. Machine Learning Methods in the Environmental Sciences. Cambridge University Press, 364 pp.
- JONES, N., 2017. How machine learning could help to improve climate forecasts. *Nature*, 379-380.